

# FTL Modula-2

**CP/M+**

**£** 54.95  
+VAT

## FTL Modula-2 *Your next Language from HiSoft*

Face it, programming can be a chore or a joy. Some computer languages bog you down in petty details, some don't help you enough. Is there any one which is balanced?

### **YES**

FTL Modula-II makes programming fun yet fast. It's easy to learn, fast to use, yet very powerful. It makes small, fast, native code files.

*Why work harder than you have to?*

### **Why Modula-2?**

It was invented by Niklaus Wirth to perfect the flaws in Pascal. ***And it does!***

### **Why FTL?**

A quick answer: speed, convenience and versatility. It compiles over 500 lines a minute, handles complicated real-world programs, knows how to search your disks for its files. This is a professional product, not a toy. It's understandable, not daunting. FTL lets you think about the program, not the compiler!

### **Complete compiler.**

The full language, not a subset! Very efficient native code, too--up to four times more compact than competing compilers.

### **Great editor.**

Edit up to three files and switch between them anytime. Run the compiler, jump right back to errors. Define and execute macros. Edits any ASCII file, too!

### **200 page manual.**

A comprehensive manual covering everything from compiler use to editing comes with every copy of FTL.

### **CP/M access.**

Get a directory, do a BIOS call, read an I/O port.

### **Assembler.**

For really low-level operations, write an assembly module, link it in your own programs (The assembler isn't necessary for writing Modula).

### **Excellent linker.**

Intelligent, disk-based, fast. Automatically includes modules you need. Tells you if something needs recompiling.

### **Not copy protected.**

We trust our customers.

### **Library Source Code included.**

Many other companies don't even offer their libraries, or do so at high cost. Ours is free.

### **Real number support, standard.**

Over 15 digits in accuracy, with accurate math and transcendentals.

# FTL Modula-2

## **Supported and improved.**

Enhancements are constantly added to FTL--CP/M and MS-DOS. Updates are reasonable so you won't go broke staying current.

## **Easy to learn, speed to burn**

Can a beginner learn it? Yes! While FTL isn't a tutorial, there are plenty of examples. The manual is written in English, not some obscure dialect of Urdhu. The section in the manual on errors is large and helpful too; no explanations of *fix and try again*.

FTL has its own editor, written in Modula-2. It will edit up to three files at once--a tremendous plus. FTL encourages programs to be written in small pieces or modules--then compiled separately. No more finding one lousy error in line 3,002 and starting over. Once compiled, a module can be used forever. Modules naturally isolate bugs, too.

If you know Pascal, it's not hard to learn Modula-2. Niklaus Wirth put the years of experience since he wrote Pascal into Modula-II, fixing most objections to Pascal. There are over a dozen books on programming in Modula, meaning lots of learning material.

Modula-2's strenghts are manifold. It will talk to hardware, but it's not hard to learn. It uses English keywords instead of mathematical symbols. Because it's not a large language, the compiler is moved easily to different

computers. Programs written in Modula can quickly move to new machines, without change.

## ***One price, one great product***

At last, all you need to start programming is right in one place. With FTL and a textbook, it's a breeze to learn Modula-2. Once you do, your programs will be portable from CP/M up to VAX. Move your programs to new computers with great ease, too, since Modula-2 is a standard language.

## **Is switching to Modula worth it?**

***Yes! Here's why:***

### **Cuts the development cycle.**

Instead of re-inventing, re-use already written code. Write a module, compile it once, use it forever. Or use someone else's library.

### **Less recompilation.**

Separate compilation is built into Modula, not "bolted on". FTL checks variables between modules, too.

### **Fewer mysterious bugs.**

Many languages, like C, don't check types between compiled files. This leads to many mysterious and confusing errors, often taking days to find. Modula always checks.

### **Support.**

We'll be here. FTL Modula-2 has been constantly updated since its first release a year ago.

# FTL Modula-2

## **Standard language.**

Unlike Pascal, standard Modula is powerful without extensions. The same program runs from Z80s through VAX. Spend time writing programs, not converting them.

## **Makes team projects easier.**

Modula encourages each team member to write an interface before any internal code. If it's not in the interface, it can't be changed by other modules.

## **Available for CP/M & MSDOS.**

No other Modula is. Write programs for Z80-based controllers and ROM-based systems or for general MSDOS use. Control interrupts and low-level I/O from Modula or write an assembly module.

*FTL Modula-2 is a complete, one-pass compiler. No language features have been left out. Some of the highlights:*

## **Tight code.**

FTL Modula-2 crafts fast, small native code. Real number code is included only if needed. Minimum program size is under 2K for CP/M, 6K for MSDOS.

## **Accurate real numbers.**

Reals are accurate to 15 places and so is the maths. Many Pascals have only 7 digit accuracy. 15 places means great accuracy for business and scientific programming. Real number support is standard, too: nothing else to buy!

## **An outstanding manual.**

Written with knowledge and wit, its 200 pages cover language features, the editor, the linker, the assembler and machine considerations. There are 15 pages alone on compilation errors and what they mean.

## **Large sets.**

Sets in FTL can have 1,024 elements, not the usual sixteen.

## **Extra-long identifiers.**

Variable names can be up to 32 characters long.

## **Incredible variety of source code.**

Source to the standard modules is provided so you can modify it yourself. It's also a pool of example code for learning the language.

## **Extra editor features in MS-DOS.**

Change directories, run programs, syntax-check programs, link, recover from disk errors, get a directory--without leaving.

## **Why Modula-2 is easier to write and maintain**

Modula syntax tames most of Pascal's hangups. IF statements look like this:

Modula IF	Pascal IF
IF Name = ID THEN	IF Name=ID THEN
	BEGIN
Error (ID);	Error (ID);
EXIT;	GOTO 999;
	END
	ELSE
ELSIF Name>ID THEN	IF Name<ID THEN
CheckRLink;	CheckRLink;
END (* IF*);	

Immediately you see the difference: no more inserting a BEGIN when adding statements to an IF. IFs are easier to read and write. And the CASE statement is much better as you will see from the example on the next page:

# FTL Modula-2

## Modula CASE

```
CASE j OF
0..9:
    DEC(1, 10);
    QT:= FALSE; |
10,11:
    Stop:= TRUE |
200: TuneBackEnd |
52:
    DecipherCase;
    QT := TRUE
ELSE HALT
END (* CASE*);
```

## Pascal CASE

```
CASE j OF
0,1,2,3,4,5,6,
7,8,9: BEGIN
    i:= 1-10;
    QT:= FALSE;
END
10,11:
    Stop:=TRUE;
200: TuneBackEnd
52: BEGIN
    DecipherCase;
    QT:= TRUE;
END;
(* Can't Do! *)
END (* CASE *);
```

Notice how each CASE is easily extended: just add code, without BEGINs and ENDs! Each CASE is separated by the vertical bar character (|), and more can be added anywhere. Modula adds an ELSE to the CASE, making it much more powerful.

### *Why Switch* *Some very important reasons:*

#### **Separate compilation.**

In standard Pascal, an entire program must be recompiled after every change. In Modula, you only compile what you've changed. FTL Modula also warns if something needs recompilation; if, for instance, you've declared a new variable.

#### **Fast compilation.**

FTL is no slouch in speed, either. It's faster to disc than Turbo Pascal 3.0.

#### **Standard means portable..**

All Pascals add features in a haphazard way. Modula is standard so code can move to new machines.

## **THE CARDINAL TYPE.**

How many times have you wanted to count past 32,000 and had to use a real number? Modula's CARDINAL type goes from 0 to 65,535 and is standard.

## **LONG Types.**

MSDOS FTL has LONGCARD and LONGINT types which count up to 4,294,967,296. Standard, not an extra.

## **Better FOR loops.**

Modula's count by any increment, not just one.

## **Opaque types.**

Write a file handler whose data can be used but not modified by other modules.

## **System access, standard.**

Get at DOS without assembly or special tricks. Or use FTL's own assembler if you like.

## *Technical improvements*

Here's some more advanced reasons to buy FTL Modula-2:

### **The compiler is one-pass,**

directly generating native code. It's faster. The compiler is of unique design, not based on any older, slower compiler.

### **The disk-based linker.**

will make large .COM files independent of available memory.

# FTL Modula-2

## **Use assembly just like Modula-2.**

Assembly code is isolated in modules, with a definition, exactly as if you'd written in Modula. Call assembly routines just like Modula, too.

## **ROMable code support.**

Move data and code areas (starting addresses) independently to anywhere in memory. Set code exit location. Programs can easily be written for variable size TPAs (CP/M).

## **Procedures can be passed as parameters.**

Use FTL's sort routine on ANY data without writing your own. Write routines which work on any data type without rewriting!

## **Open array parameters.**

In Pascal, an array passed to a procedure must be fixed in length; FTL can pass any size array and its length. Open parameters can be VAR or constant.

## **Access to interrupts.**

Use Z80 mode 2 interrupts.

## **Assembler included.**

Assembly code may be linked into your programs. Assembly is not necessary for writing Modula-2.

## **Very accurate REAL numbers.**

Real numbers are accurate to over 15 places and so is FTL's maths.

## **Fast, small integer maths.**

If you need top speed then the integer maths comes into its own. See the PCW Benchmarks in this catalogue.

## **FTL Editor/ToolKit**

£ 34.75 + VAT

The Editor/ToolKit contains full source to the editor in FTL Modula-2; the editor is written in Modula-2. It's a fast, macro editor which will edit up to three files at once, remembers where files were last edited, calls the compiler and returns.

The modules on this disk are valuable Modula-2 tools. Simply compile them and use them. Programs you write using the Editor/ToolKit are yours. The editor source code itself is copyrighted and may not be sold.

Disk contents include: comment file on the Editor/toolkit disk, contents, and files; CAT, CATenate program: displays files to screen or disk file; COMMAND, processes CP/M command line; COMPARE, compares text differences; COMPDIR, compares entire directories of files; DOMENU and control, display, and status modules; KEYBOARD, MACROS, keyboard I/O and keyboard macro control; MLU, library mangement utility; PATTERN, pattern-matching routine; SCREENIO, does screen I/O, GOTOXY, highlighting, line drawing, etc.

FTL Modula-2 comes with full support from HiSoft at a remarkable price.

In addition to CP/M FTL Modula-2 we have PC (PC1512, PC1640 & compatibles) and Atari ST versions of the package so that portable development is now a reality.

**All you need is FTL Modula-2.**



# Benchmarks

The following is a collection of benchmark timings for some of the programs listed in this catalogue. It is not meant to be complete but should give an indication of the performance of the packages.

The benchmarks are based on the Personal Computer World tests (for Pascal, Modula-2 and C) and the Byte Sieve test (for ZBASIC).

Package	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Pascal80	0.2	3.8	6.2	5.3	4.8	5.1	25.3	20.3	11.6	8.5	4.3	8.5	5.3	5.2	8.6
Aztec C	0.5	6.3	6.4	5.5	7.7	7.6	14.0	15.6	11.6	13.2	5.7	11.8	6.6	6.6	52
HiSoft C	1.3	15.5	17.0	15.5	17.1	17.8	***	***	99.1	24.5	6.0	24.8	6.7	6.7	***
Modula-2	.03	4.2	6.7	6.0	5.2	5.4	192	164	10.5	9.3	1.02	9.2	3.0	3.1	265

Timings in seconds, HiSoft C is integer only and thus has no timings for the float benchmarks (7,8 & 15). FTL Modula-2 is fairly slow on floats but *very* accurate.

Package	Sieve Benchmark
ZBASIC	3.5
Mallard BASIC	133.0

Timings in seconds for one iteration.